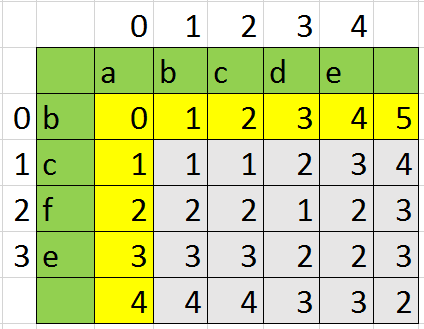
**Edit Distance**



In order to find the number of changes in str1 it is better to use similar algorithm with Longest common sub sequence, but with some differentions. We will use matrix LS[][]. Lets first initialize LP[i][0]=i and LP[0][i] where 0<i<n. if str1[i-1]==str[j-1] then LP[i][j]=LP[i-1][j-1], because LP[i-1][j-1] saves changes on the part of the string str1 with length i and the part of the string str2 with length j. Otherwise, we will look at minimum of LP[i-1][j-1], LP[i][j-1] and LP[i][j-1] that will give us minimum changes on the different lengths of our strings.

The algorithm will take quadratics memory and run time ( O(n\*m)).